

Feature Extraction of Volume Data Based on Multi-Scale Representation

Y. Wu, E.-C., Chang, Z. Huang, M. S. Kankanhalli
Department of Computer Science
School of Computing
National University of Singapore
3 Science Drive 2, Singapore 117543

Abstract

In this paper, we present a novel algorithm that can detect and extract *salient points* as features in 3-D volume datasets. This algorithm extracts not only the locations of feature points, but also finds out the scales at which the features are most significant. It applies the scale-space theory by adaptively processing the input volume in a few discrete scales. The features points, as well as their scales, detected can be used for subsequent processing, such as content-based volume data authentication and content-based volume rendering.

We have implemented the algorithm and tested its effectiveness on several volume datasets. Some optimization has also been done on time-consuming intermediate steps to speed up the feature detection process.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling; I.4.6 [Image Processing]: Segmentation; I.5.4 [Compute Vision]: Applications

Keywords: Medical imaging, Scientific visualization, Feature detection, Volume processing, Scale-space, Multi-scale, Applications

1 Introduction

Feature detection in 3-D volume data is important for many applications such as content-based volume data authentication and content-based volume rendering. However, it is not as intuitive as that in 2-D image, since there is no specific definition on what a feature is in the context of volume data. Aiming to design a technique as a pre-processing step for other volume-based applications, we propose a volume feature extraction method based on the scale-space representation. We define features, or 3-D *salient points*, as grey-level local extrema. Intuitively, grey-level local extrema in a volume dataset are blob-like structures which specify both the location and the size (or *scale*) of each feature point.

Although feature detection based on surface structures can be useful to applications like surface reconstruction, it is not so applicable to some content-based techniques. The major problems are:

1. Features in 3-D volume data are not as commonly agreed upon as those in 2-D image data. It is relatively unclear what *surfaces* really are when the content of a volume dataset is noisy.

Isosurfaces in volume data are intuitive, but they do not necessarily capture the actual content.

2. Some algorithms for 2-D cases can be computationally expensive, which will become a major concern when ported directly to 3-D cases. This is because data sizes of volume data are usually much larger than those of images.
3. In terms of feature detection, we may want not only features with high intensity values, which are similar to edges or textures in the 2-D case, but also parts with very low intensity values (or *holes*). In this context, describing holes with the surrounding surface may not seem intuitive, especially for volume data visualization, etc.

We have implemented the algorithm and tested its effectiveness on several volume datasets. Some optimization has also been done on time-consuming intermediate steps to speed up the feature detection process.

The remaining part of the paper is organized as follows: First, in Section 2, we discuss the multi-scale representation of N -D signal. It is a generalization from the representation of the 1-D signal. Second, in Section 3, we describe how to approximate 3-D Gaussian kernels for efficient computing. Third, in Section 4, we extend the notion of salient points to 3-D volumes. Next, in Section 5, we propose the algorithm to extract feature points of 3-D volumes. Then, in Section 6, we show the results from our implementation. In Section 7, we brief the efforts of the related work before concluding the paper in Section 8.

2 Multi-Scale Representation

The scale-space theory was originated from an analogue of how we perceive the objects around us. It is intuitive that objects in the world and details in images only exist as meaningful entities over a limited ranges of scale. This is oppose to the ideal mathematical entities like “points,” “lines,” etc., which appear exactly the same regardless the scales of observation.

The main idea of creating a multi-scale representation of a signal is by generating a one-parameter family of derived signals from the original signal. As illustrated in Figure 1, details are suppressed successively from the original signal to coarser levels of scales. Thus, the key here is to define a mechanism that can systematically simplify the original signal and remove finer-scale details, which should be available at any level of scale.

Scale-space for N -D signals: Following the formulation for 1-D continuous signal[Lindeberg 1994], we can generalize the definition to a framework for N -D continuous signals:

Given a signal $f : \mathbb{R}^N \rightarrow \mathbb{R}$, the scale-space representation $L : \mathbb{R}^N \times \mathbb{R}_+ \rightarrow \mathbb{R}$ is defined such that the representation at zero scale is equal to the original signal

$$L(\cdot; 0) = f, \quad (1)$$

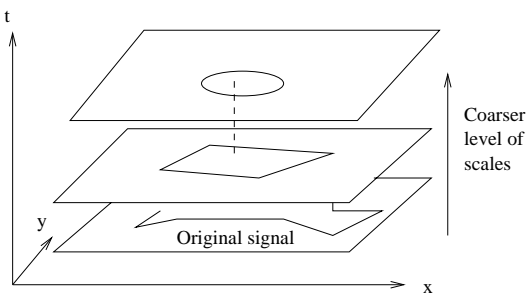


Figure 1: An illustration of the multi-scale representation of a 2-D signal.

and the representation at coarser scales is given by convolution with Gaussian kernels of increasing width,

$$L(\cdot; s) = g(\cdot; s) * f. \quad (2)$$

In terms of explicit integrals, the result of the convolution operation ‘*’ is written

$$L(x; s) = \int_{\xi \in \mathbb{R}^N} g(\xi; s) f(x - \xi) d\xi, \quad (3)$$

where $x = (x_1, \dots, x_N)^T \in \mathbb{R}^N$, and $g : \mathbb{R}^N \times \mathbb{R}_+ \setminus \{0\} \rightarrow \mathbb{R}$ is the N -D Gaussian kernel of standard deviation $\sigma = \sqrt{s}$

$$g(x; s) = \frac{1}{(2\pi s)^{N/2}} e^{-(x^T x)/2s}. \quad (4)$$

Let’s illustrate multi-scale representation of a 2-D image using Gaussian kernels of different width. The results are shown in Figure 2. Note that the finer-scale details of the mandrill face is successively suppressed, such that the eyes, the bright parts of the face, and the dark parts of the cheeks become more and more obvious.

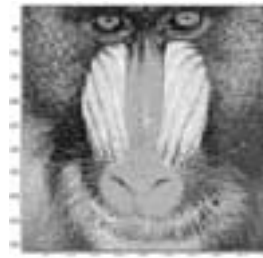
3 Approximating a sequence of 3-D Gaussian Kernels

To find the feature in the scale-space, we need to first compute the scale-space representation at various discrete scales, s_0, s_1, \dots, s_d . Usually, the scales is in the form of $s_i = s^i$ where $s > 1$ is some predefined constant.

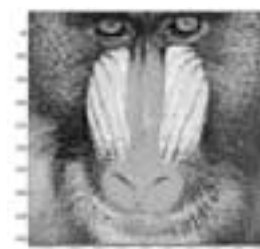
To compute these scales directly from the definition is computational intensive, especially in 3-D. The time complexity of a direct computation of just one scale is at least $\Omega(v^3 k^3)$, where v^3 and k^3 is the size the volume (in term of number of voxels) and the kernel respectively. Because the support of the Gaussian is infinite, the size of the kernel should be v . Although the kernel at scale s can be well-approximated by a box of width $10\sqrt{s}$, even in this reduced size, the computation is still intensive.

It is possible to speedup computation by approximating the kernel in another way. For example, [Rau and McClellan 1997] extends a well-known approximation in 1-D to 2-D and give the relationship between the computing time (in term of number of arithmetic operations) and approximation accuracy. Generalization the idea to 3-D is relatively straight-forward.

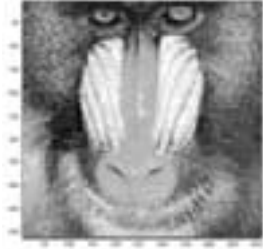
In our problem, instead one single scale, we are required to compute multiples scales. Thus, using the approximation independently in each scale might repeated certain computation. In this implementation, given the required scales, we study them carefully and design a sequence of approximations that share intermediate computation across the scales. In other words, we try to optimize the overall cost, rather than the cost in each scale.



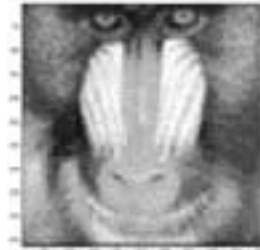
(a) The original of the mandrill face (512 × 512).



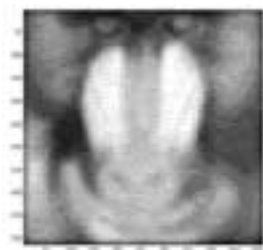
(b) The mandrill face convolved with Gaussian kernels of width 4.



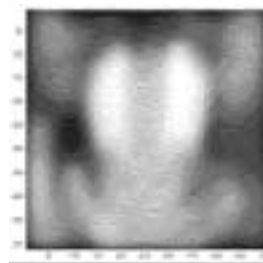
(c) The mandrill face convolved with Gaussian kernels of width 16.



(d) The mandrill face convolved with Gaussian kernels of width 32.



(e) The mandrill face convolved with Gaussian kernels of width 64.



(f) The mandrill face convolved with Gaussian kernels of width 128.

Figure 2: Successive smoothing of a grey-level image (mandrill face) with 2-D Gaussian kernels.

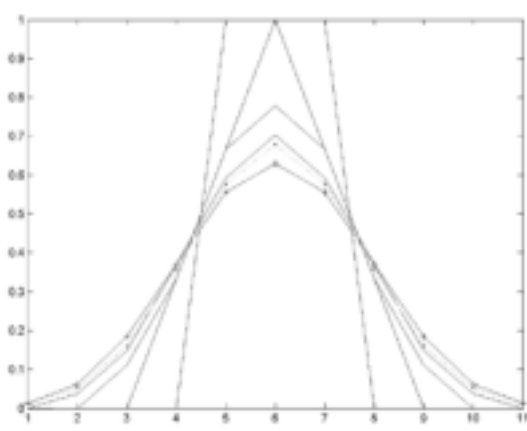


Figure 3: Approximating 1-D Gaussian with a series of convolutions with rectangular kernel.

To illustrate our method, let us start from the 1-D case. It is well-known that a Gaussian kernel at a particular scale can be approximated by a spline function, which can be obtained by convoluting a size m rectangular kernel k number of times. A rectangular kernel of size m consists of m consecutive 1's. Let denote this filter as $[1]_m$, and the spline function as $[1]_m^k$. Note that there are many possible pairs of m and k that satisfy a given accuracy. As shown in Figure 3, the dotted line is the sampled 1-D Gaussian function with sample points marked by black dots. The other solid lines are approximated functions with a series convolutions with a rectangular kernel. After 3 convolutions, we can observe that the 1-D function already approximates the sampled Gaussian function with very high degree of accuracy.

It is also known that convoluting with a rectangular kernel $[1]_m$ can be done very efficiently, in fact, in only 2 arithmetic operation (one addition and one subtraction) per sample. Interestingly, the amount of operations required is independent of the size m . Note that repeatedly applying the rectangular kernel k times is equivalent to convoluting with $[1]_m^k$, which required only $2k$ arithmetic operations. Given the scale s and required approximation accuracy, we can determined the optimal k and the corresponding m .

Similarly, we can approximate 3-D Gaussian kernels by a series of convolutions with “box kernels” of different sizes, which consist of a cube of 1's in the center and 0's elsewhere.

Now, given the required scale s_0, s_1, \dots, s_d , a possible speedup is to treat each scale individually. However, we can obtain further speedup by combining intermediate results. For example, if, for s_0 and s_3 , we choose the approximation $[1]_5^4$ and $[1]_5^7$, then the number of operation is only 2×7 per sample. This is an improvement from computing both separately, which required $2 \times 4 + 2 \times 7$ operations. Although $[1]_5^4$ and $[1]_5^7$ may not be optimal in each scale, the overall cost maybe better than combining the optimal cost in each individual scale. In current implementation, we studied the given scales and carefully select the box filters that share intermediate results. It would be interesting to give an algorithm that compute the overall optimal solution.

Special care must be taken in implementation. We further optimize the way the array indices are accessed in the program, which reduce index computations as well as buffering time.

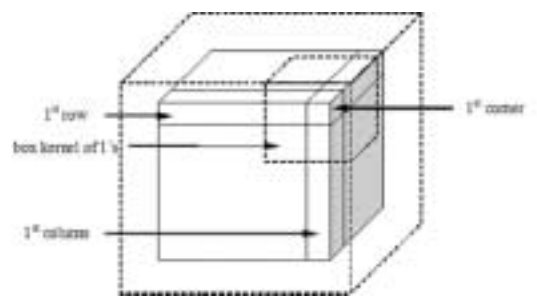


Figure 4: Illustration of the fast box convolution optimization.

4 Volume Features as Scale-Space Salient Points

Similar to the definition of salient points in the 2-D images [Lindeberg 1990], we extend the notion of salient points to 3-D volumes. The scale-space representation $L(x; s)$ for a volume is determined by a kernel $g : \mathbb{R}^3 \rightarrow \mathbb{R}$. A volume at scale s , $L(\cdot; t)$ is the convolution of the original volume with the dilated kernel $g_s = (1/s)g(\cdot/\sqrt{s})$. By applying Gaussian kernel here, we have a special case of the N -D equation

$$\begin{aligned} L(x; s) &= \{f * g_s\}(x) \\ &= \frac{1}{(2\pi s)^{3/2}} \int_{\xi \in \mathbb{R}^3} f(x - \xi) e^{-\|x\|_2^2/2s} d\xi. \end{aligned} \quad (5)$$

5 Feature Detection in Volumes

The 3-D feature detection and extraction algorithm is a direct extension from the 2-D counterpart from higher-level point of view. While they look the same in the first glance, in the implementation details, however, the 3-D case does require more considerations and slightly different implementation than the original 2-D case.

The basic algorithm for volume feature detection is outlined as follows:

1. Given a volume dataset V ,
2. for $s = s_0$ to s_1 ,
 - Convolve V with a Gaussian-like kernel specified by scale s ;
 - Keep each smoothed volume V_s ;
 - end for;
3. Look for local maxima in both the spatial domain in each V_s and in the scale domain across adjacent scales;
4. Combine points of local maxima that are too *close* to each other;
5. Sort all feature points in order of their strengths, and pick the strongest n points as output.

6 Experiment Results

Now, we show the feature extraction results using the scale-space method. We first show the results of a 2D example (Figure 5). The locations of the salient points are the centers of the circles; the scales t are loosely represented by the radii of the circles; the absolute values of strengths $|m|$ are loosely represented by the widths of the circles' borders. Note that since only the strongest 30 salient

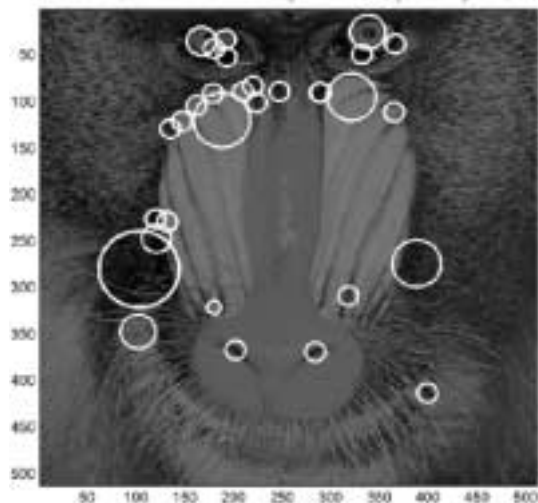


Figure 5: Scale-space salient points superimposed onto a dimmed version of the original mandrill face.

points are shown, their strengths do not differ much from one another.

Similar to 2D image, we show the extracting results of 3D volume data. They are shown in Figures 6, 7, 8 and 9. Brighter parts in the images denote areas with larger grey-level voxel values. Note that how the salient points match portions of the original volume where grey-level voxel values differ from the surrounding area. In Figures 7 and 9, the locations of the salient points are the centers of the circles; the scales t are loosely represented by the radii of the circles; the absolute values of strengths $|m|$ are represented proportionally by the widths of the circles' borders (dotted line denote positive strength, solid line denote negative strength). Note that only the strongest 15 features are shown.

7 Related Work

Many ideas from 2-D cases are applicable for and easy to extend to handle their 3-D counterparts. Thus, it is intuitive to extend *edges* in 2-D cases to *edges* in 3-D cases which turn out to be *surfaces* [Audette et al. 1999].

It is not surprise to see that many proposed feature detection techniques aim to extract surface structures in volume data, and attempt to perform segmentation of the input volume based on surfaces extracted. Most common techniques in 2-D cases, such as Laplacian of Gaussian [Bomans et al. 1990], non-maxima suppression [Pratt 1991], functional fitting [Morgenthaler and Rosenfeld 1981] have been ported to handle 3-D cases.

Although the implementation is easily approximated in discrete domain like volume data, it bears a few disadvantages:

1. Threshold values and the width of the masks used in convolution must be manually chosen. This is a problem common to all gradient-based methods since both the threshold value and the width of the convolution mask are highly sensitive to any change of the width of the gradients in the input data.
2. Corners are often missed because the gradient values at corners are usually relatively small. This problem imposes difficulties for applications such as junction detection.
3. Odd symmetric filters, for example, a first derivative operator, can find only the step-like features, which actually misses

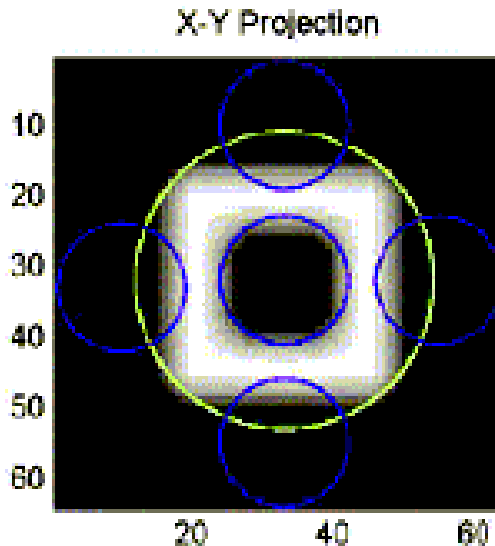


Figure 6: Feature points detected in a volume dataset with a rectangular cube. Note that all of the tunnel in the center, the four walls of the tube and the tube's body are circled as features.

ridge-like structures (so-called bar features). This can be solved by applying an even symmetric filter and look for peaks, instead of zero-crossings, from the output.

Therefore, we can observe that classical differential edge feature detection schemes suffer from both false positives and false negatives.

8 Conclusion and Future Work

In this paper, a feature detection technique for 3-D volume data was proposed. Using our method, both the position and the strength of feature points can be derived.

We are working on a framework of content-based volume data authentication. Our method uses scale-space representation for automatic feature extraction, as reported here, together with wavelet-based foveation [Chang et al. 2000], and encryption to achieve a good performance. We are still refining the framework and conducting the experiment study.

9 Acknowledgment

We would like to thank the reviewers of Graphite 2003. Their comments have been used to improve this paper. The work was partly supported by NUS scholarship for undergraduate study and the research grand R252-000-090-112.

References

- AUDETTE, M., FERRIE, F., AND PETERS, T. 1999. *An Algorithmic Overview of Surface Registration Techniques for Medical Imaging*. *Medical Image Analysis*. Oxford University Press.
- BOMANS, M., HOHNE, K.-H., TIEDE, U., AND RIEMER, M. 1990. 3-d segmentation of mr images of the head for 3-d display. *IEEE Trans. Medical Imaging* 9, 2, 177-183.

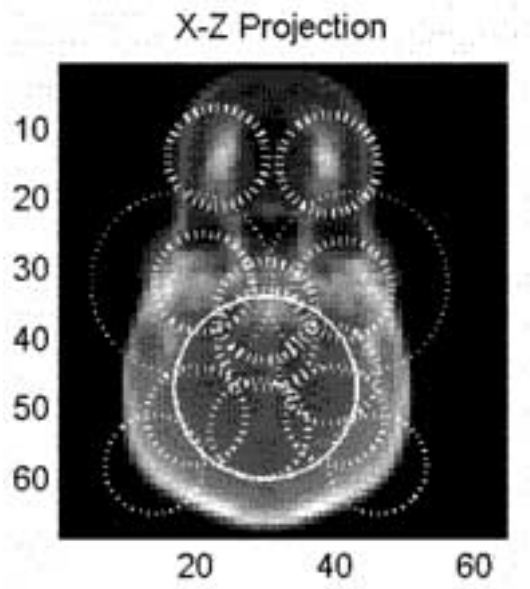


Figure 7: Feature points detected in a human skull dataset.

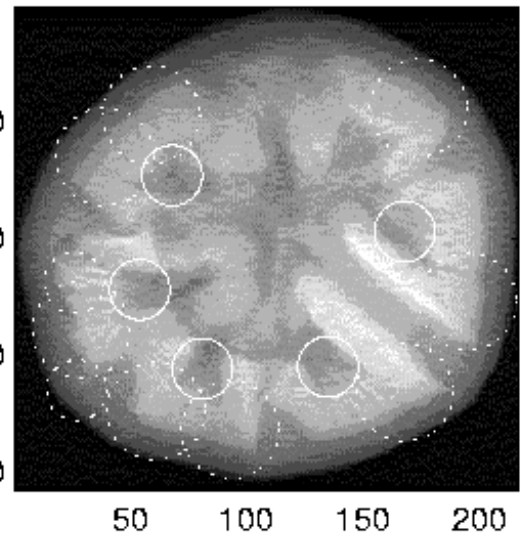
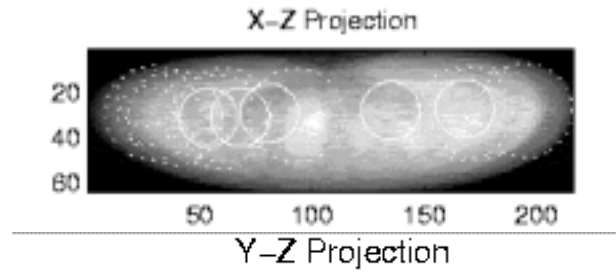


Figure 9: Feature points detected in a tomato dataset. Projections along two axes are presented.



Figure 8: Feature points detected in a human skull dataset. Scales and strengths of the salient points, however, are not shown in this rendering.

- CHANG, E.-C., MALLAT, S., AND YAP, C. 2000. Wavelet foveation. *Journal of Applied and Computational Harmonic Analysis* 9, 3, 312–335.
- LINDBERG, T. 1990. Scale-space for discrete signals. *IEEE Trans. Pattern Analysis and machine Intelligence* 12, 3, 234–254.
- LINDBERG, T. 1994. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, Netherlands.
- MORGENTHALER, D. G., AND ROSENFELD, A. 1981. Multidimensional edge detection by hypersurface fitting. *IEEE Trans. Pattern Analysis and Machine Intelligence* 3, 4, 482–486.
- PRATT, W. 1991. *Digital Image Processing*. Wiley, New York.
- RAU, R., AND MCCLELLAN, J. 1997. Efficient approximation of gaussian filters. *IEEE trans. Signal Processing* 45, 2, 468–471.